Data Mining and Machine Learning

Project Proposal: Sentiment Analysis on YouTube Comments

Team Members

- Muhammad Ali Raza (BSDSF22M011)
- Mohammad Ammar Siddique (BSDSF22M041)
- Ali Hamza (BSDSF22M032)

1. Problem Statement & Motivation

User-generated comments on YouTube videos contain rich signals about viewer satisfaction, criticism, and emerging trends. Automatically classifying comment sentiment (positive, neutral, negative) can help:

• Content creators monitor reception and tailor future videos.
• Moderators filter toxicity or spam.
• Marketers gauge brand engagement and campaign impact.

Despite many general-purpose sentiment tools, few focus on the idiosyncrasies of YouTube comments (slang, emojis, thread structure). We propose an end-to-end web application that (a) classifies individual comments

(b) ingests any YouTube video URL to produce an aggregate sentiment report.

2. Objectives & Performance Targets

• Build a model to assign each comment one of {positive, neutral, negative}.
• Implement a service that fetches comments from a provided video URL and visualizes sentiment distributions.
• Performance targets:
– ≥ 85% overall accuracy
– ≥ 80% F1-score on the “negative” class.

3. Dataset Description

Dataset: YouTube Comments Dataset (Atif Ali, Kaggle)
Link: https://www.kaggle.com/datasets/atifaliak/youtube-comments-dataset
Size: ~78000 comments ; 80% for training, 20% holdout for testing.

4. Proposed Methodology

Data Preprocessing: Clean HTML/URLs/emojis, tokenize, remove stop-words, lemmatize.
Feature Engineering: TF-IDF vectors, optional GloVe embeddings, PCA for dimensionality reduction.
Modeling: Naïve Bayes baseline; optional Logistic Regression or Random Forest; hyperparameter tuning via grid search.
Validation: 80/20 split, K-fold cross-validation; report accuracy, precision, recall, F1-score.
Application: Python backend (FastAPI/Django), YouTube Data API integration, minimal frontend/CLI.

5.Timeline (2 Weeks)

Week	Activities
Week 1	• Data prep & exploration: cleaning, TF-IDF, PCA. • Baseline modeling with Naïve Bayes, 5-fold CV, metric evaluation.
Week 2	• Backend & API: implement classify and analyze endpoints. • YouTube API integration, frontend/CLI development. • Testing, optimization, reporting, and deliverable preparation.

6. Feasibility & Originality

• Feasibility: Clear scope, proven NLP methods, and available Python tooling enable completion in two weeks.
• Originality: Real-time URL-driven analysis and PCA-accelerated classification offer unique value compared to existing open-source tools.